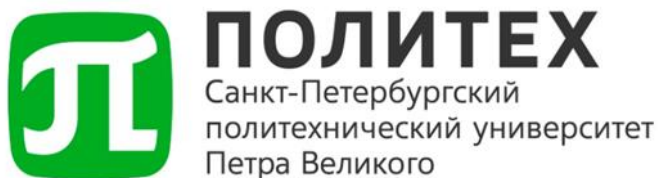


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА  
ВЕЛИКОГО»

Институт компьютерных наук и технологий

**Высшая школа программной инженерии**



**КУРСОВОЙ ПРОЕКТ**

*Автоматизация работы библиотеки*

по дисциплине «Системы управления базами данных»

Студент гр. 3530202/70201

*И.Х.Имхасина*

Руководитель, доцент, к.т.н.

О.В. Прокофьев

Санкт-Петербург

2020 г

Оглавление	
Введение .....	3
Требования.....	4
Схема базы данных .....	5
Описание таблиц .....	6
Используемые запросы.....	7
Описание и внешний вид клиента .....	11
Вывод .....	15
Список использованной литературы .....	16

## **Введение**

В рамках данного цикла лабораторных работ необходимо автоматизировать работу библиотеки. Для этого в рамках базы данных ORACLE необходимо создать объекты в схеме вашего пользователя (логин и пароль пользователя для доступа к базе данных студент должен получить у преподавателя) и написать клиентское приложение на базе компонентов ADO.NET или JDBC.

Библиотека является государственным предприятием и выдает литературу жителям Выборгского района (далее Клиентам) на безвозмездной основе. Клиенты характеризуются фамилией, именем и номером паспорта. Книги библиотеки характеризуются полным наименованием и разбиты на 3 категории: обычные, редкие и уникальные. Экземпляров книг в библиотеке ограниченное количество и при выдаче и возврате их необходимо вести учет числа экземпляров каждой книги. Библиотекари при выдаче и приеме книг обязаны учитывать все операции в журнале. При выдаче книги в журнал необходимо заносить дату выдачи (автоматически: текущая дата), клиента (библиотекарь выбирает из справочника), книгу (библиотекарь выбирает из справочника) и максимальную дату возврата (автоматически: текущая дата + количество дней, в зависимости от типа книги). При приеме книги устанавливается реальная дата возврата и, при необходимости, исчисляется штраф.

## **Требования**

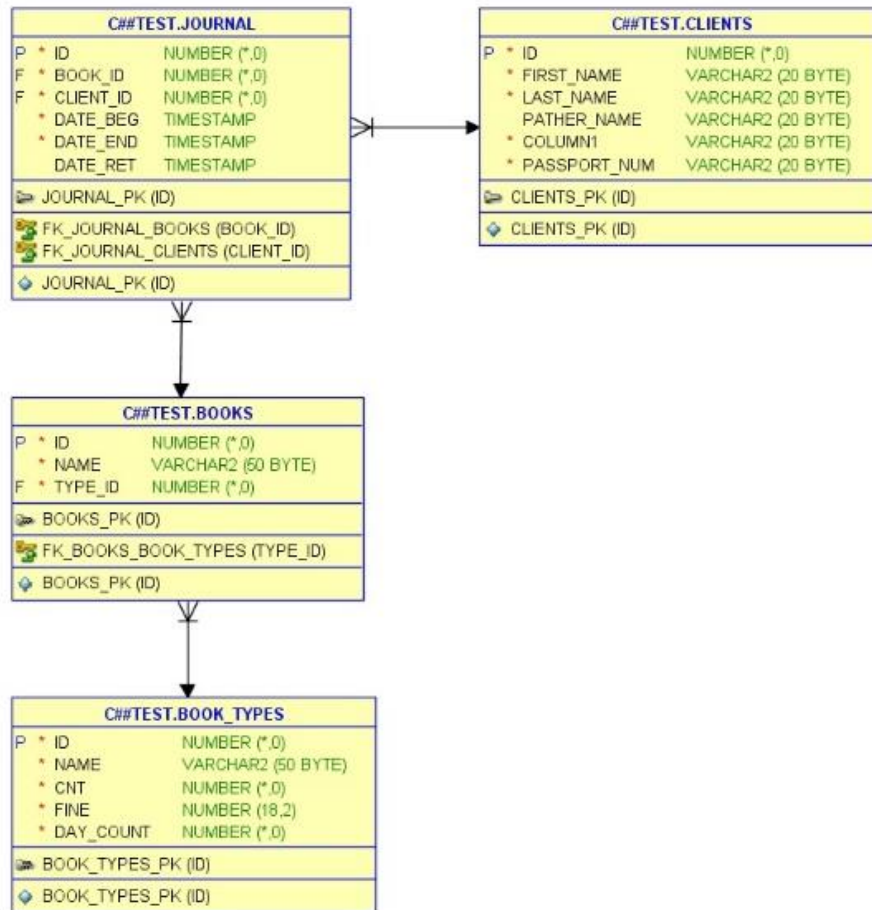
**База данных должна удовлетворять следующим требованиям:**

1. Контроль целостности данных, используя механизм связей.
2. Операции модификации групп данных и данных в связанных таблицах должны быть выполнены в рамках транзакций.
3. Логика работы приложения должна контролироваться триггерами. В частности:
  - Триггер должен не позволять выдать книгу, не имеющуюся в наличии.
  - Триггер должен контролировать, что выдаваемая книга выдается клиенту, имеющему на руках менее 10 книг.
4. Все операции вычисления различных показателей (из требований к клиентскому приложению) должны реализовываться хранимыми процедурами.

**Требования к клиентскому приложению:**

1. Необходимо реализовать интерфейсы для ввода, модификации и удаления:
  - Клиентов;
  - Книг;
  - Типов книг.
2. В главном окне приложения должен быть реализован журнал библиотекаря с возможностью выдачи и приема книг.
3. Необходимо реализовать возможность просмотра библиотекарем следующих показателей:
  - Число книг на руках у клиента.
  - Размер самого большого штрафа.
  - Размер штрафа данного клиента.
  - Три самые популярные книги.

## Схема базы данных



### Описание таблиц

Имя таблицы	Расшифровка
clients	Читатели: Id, фамилия, имя, паспорт
books	Книги : Id, наименование, тип (фк), количество
book_types	Типы книг : Id, название, штраф, число дней
journal	Журнал библиотекаря: Id, Клиент(фк), Книга(фк), Даты взятия, возврата и возвращения, штраф

Связи между таблицами:

Название	Primary Key	Foreign Key
fk_journal_clients	clients.id	journal.client_id
fk_journal_books	books.id	journal.book_id
fk_books_book_types	book_types.id	books.type_id

## Используемые запросы

- Вход в приложение

```
Select count(*) from rulers where name = ' login' and password=(select DBMS_CRYPTO.HASH(rawtohex('password'),2) as md5 from dual)
```

- Вывод таблицы журнала

```
select journal.id,CLIENT_ID,FIRST_NAME,LAST_NAME,BOOK_ID,name,trunc(date_beg) as from_,trunc(date_end) as to_,trunc(date_ret) as back_ from journal JOIN CLIENTS ON journal.CLIENT_ID=CLIENTS.ID JOIN BOOKS ON journal.BOOK_ID=BOOKS.ID
```

- Вывод таблицы клиентов

```
select * from clients
```

- Вывод таблицы книг

```
select * from clients
```

- Вывод таблицы типов книг

```
select id, name,fine,day_count from book_types
```

- Добавление клиента

```
create or replace Procedure add_client (FIRST_NAME_ IN VARCHAR2, LAST_NAME_ IN VARCHAR2, PATHER_NAME_ IN VARCHAR2, PASSPORT_SERIA_ IN VARCHAR2, PASSPORT_NUM_ IN VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
INSERT INTO CLIENTS (FIRST_NAME, LAST_NAME, PATHER_NAME, PASSPORT_SERIA, PASSPORT_NUM)  
VALUES(FIRST_NAME_, LAST_NAME_, PATHER_NAME_, PASSPORT_SERIA_, PASSPORT_NUM_);
```

```
END add_client;
```

- Обновление данных клиента

-Имя и фамилия

```
create or replace Procedure upd_client(id_ IN VARCHAR2, FIRST_NAME_ IN VARCHAR2, LAST_NAME_ IN VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
UPDATE clients SET FIRST_NAME = FIRST_NAME_ WHERE ID= id_;  
UPDATE clients SET LAST_NAME = LAST_NAME_ WHERE ID= id_;
```

```
End upd_client;
```

- серия и номер паспорта

```
create or replace Procedure upd_client_PASS(id_ IN VARCHAR2, PASSPORT_SERIA_ IN  
VARCHAR2, PASSPORT_NUM_ IN VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
UPDATE clients SET PASSPORT_SERIA = PASSPORT_SERIA_ WHERE ID= id_;
```

```
UPDATE clients SET PASSPORT_NUM = PASSPORT_NUM_ WHERE ID= id_;
```

```
End upd_client_PASS;
```

-Возврат книги

```
create or replace Procedure book_back(id_ IN VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
UPDATE JOURNAL SET DATE_RET = trunc(sysdate) WHERE ID= id_;
```

```
End book_back;
```

-удалить клиента

```
create or replace Procedure del_client(id_ IN VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
DELETE FROM journal WHERE client_ID= id_;
```

```
DELETE FROM Clients WHERE ID= id_;
```

```
End del_client;
```

-добавление книги

```
create or replace Procedure add_book (NAME_ IN VARCHAR2, cnt_ IN number,tid IN  
VARCHAR2)
```

```
IS
```

```
BEGIN
```

```
INSERT INTO BOOKS (ID, NAME,CNT,TYPE_ID)
```

```
VALUES(BOOKS_SEQ.nextval,NAME_, cnt_, tid);
```

```
END add_book;
```

-модификация количества книг

```
create or replace Procedure upd_b_count(id_ IN VARCHAR2, cnt_ in number)
```

```
IS
```

```
BEGIN
```



```
UPDATE BOOKS SET CNT =cnt_ WHERE ID=id_;
```

```
End upd_b_count;
```

-удаление книги

```
create or replace Procedure del_book(id_ IN VARCHAR2)
IS
BEGIN
```

```
DELETE FROM journal WHERE book_ID= id_;
DELETE FROM books WHERE ID= id_;
```

```
End del_book;
```

- обновление типов книг

```
UPDATE BOOK_TYPES SET FINE =newfine WHERE ID=id_;
```

- штраф клиента

```
create or replace Procedure cl_fee (id_ IN varchar2, N OUT NUMBER)
IS
Begin
```

```
SELECT sum((TRUNC DATE_RET)-TRUNC DATE_END))*BOOK_TYPES.FINE)
INTO N FROM JOURNAL INNER JOIN
CLIENTS ON JOURNAL.CLIENT_ID = CLIENTS.ID INNER JOIN
BOOKS ON JOURNAL.BOOK_ID = BOOKS.ID INNER JOIN
BOOK_TYPES ON BOOKS.TYPE_ID = BOOK_TYPES.ID GROUP BY
CLIENTS.ID,BOOKS.NAME,JOURNAL.DATE_END,JOURNAL.DATE_RET
HAVING JOURNAL.DATE_RET > JOURNAL.DATE_END AND CLIENTS.ID = id_;
```

```
End cl_fee;
```

-вывод самого большого штрафа

```
Select * From( SELECT CLIENTS.ID, CLIENTS.FIRST_NAME, CLIENTS.LAST_NAME,
(TRUNC DATE_RET) - TRUNC DATE_END)) * BOOK_TYPES.FINE AS FINAL FROM
JOURNAL
INNER JOIN CLIENTS ON JOURNAL.CLIENT_ID = CLIENTS.ID
INNER JOIN BOOKS ON JOURNAL.BOOK_ID = BOOKS.ID
INNER JOIN BOOK_TYPES ON BOOKS.TYPE_ID = BOOK_TYPES.ID
GROUP BY CLIENTS.ID,
CLIENTS.FIRST_NAME, CLIENTS.LAST_NAME,
BOOKS.NAME, BOOK_TYPES.NAME,BOOK_TYPES.FINE,JOURNAL.DATE_END
JOURNAL.DATE_RET, TRUNC DATE_RET) - TRUNC DATE_END)
HAVING JOURNAL.DATE_RET > JOURNAL.DATE_END
ORDER BY FINAL DESC) WHERE ROWNUM = '1';
```

-Число книг у клиента на руках

```
create or replace Procedure cl_have_books (id_ IN varchar2, N OUT NUMBER)
IS
Begin
```

```
Select Count(*) INTO N from journal where Date_Ret is null and client_id=id_;
```

```
End cl_have_books;
```

-топ3 популярные книги

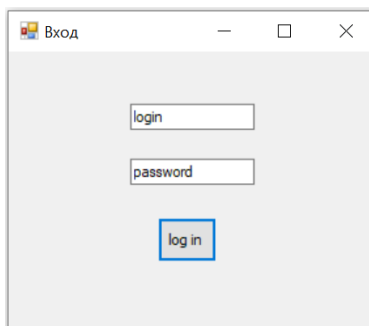
```
SELECT * FROM (SELECT BOOKS.ID, BOOKS.NAME, COUNT(JOURNAL.BOOK_ID) as
kol
FROM BOOKS INNER JOIN JOURNAL ON JOURNAL.BOOK_ID = BOOKS.ID
GROUP BY BOOKS.ID, BOOKS.NAME, BOOKS.CNT, JOURNAL.BOOK_ID ORDER BY
COUNT(JOURNAL.BOOK_ID) DESC) WHERE ROWNUM<4;
```

-Взятие книги реализовано с автоназначением дат:

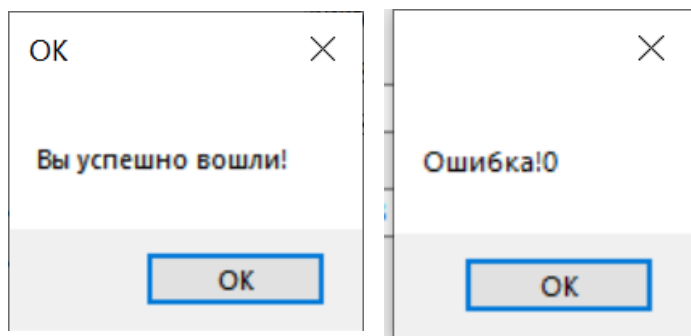
```
Select DAY_COUNT from BOOK_TYPES where ID = (SELECT TYPE_ID FROM BOOKS
WHERE id=b_id);
d=day_count/31;
INSERT INTO JOURNAL(BOOK_ID, CLIENT_ID, date_beg, date_end, date_ret)
VALUES(b_id, cl_id, SYSTIMESTAMP, ADD_MONTHS(SYSTIMESTAMP, d), NULL);
```

## Описание и внешний вид клиента

Для выполнения входа пользователю необходимо ввести логин и пароль. Введенные данные сравниваются с данными из таблицы Rulers, в которой хранятся логины и хеши паролей.

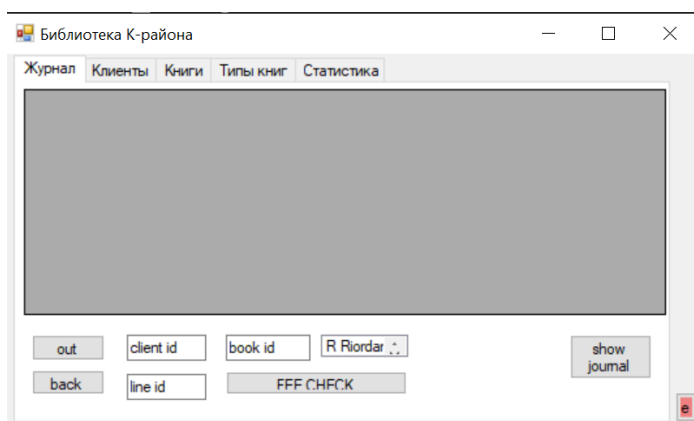


После нажатия на кнопку «log in» выводится информация об успешном входе в базу данных или о неверно введенном логине или пароле.



После входа открывается форма, содержащая несколько вкладок: Журнал, Книги, Клиенты и Статистика.

На вкладке Журнал можно посмотреть таблицу, содержащую все записи в журнале библиотекаря, также можно добавить запись о выдаче и вернуть книгу, а также проверить назначен ли штраф. Чтобы избежать проблем с читателями, у которых полностью совпадают ФИО, они вносятся по своему id, который условно считается их номером читательского билета. Книги выбираются по названию. Чтобы вернуть книгу нужно указать номер записи из журнала.



Библиотека К-района

Журнал Клиенты Книги Типы книг Статистика

	LAST_NAME	BOOK_ID	NAME	FROM_	Ti
▶	alekseeva	1	R Riordan PJ	21.05.2020	21
	alekseeva	1	R Riordan PJ	21.05.2020	21
	petrovna	1	R Riordan PJ	21.05.2020	21
	alekseeva	1	R Riordan PJ	27.02.2020	29
	petrovna	1	R Riordan PJ	22.02.2020	29

Библиотека К-района

Журнал Клиенты Книги Типы книг Статистика

	LAST_NAME	BOOK_ID	NAME	FROM_	Ti
▶	alekseeva	1	R Riordan PJ	21.05.2020	21
	alekseeva	1	R Riordan PJ	21.05.2020	21
	petrovna	1	R Riordan PJ	21.05.2020	21
	alekseeva	1	R Riordan PJ	27.02.2020	29
	petrovna	1	R Riordan PJ	22.02.2020	29

Библиотека К-района

Журнал Клиенты Книги Типы книг Статистика

	LAST_NAME	BOOK_ID	NAME	FROM_
▶	petrovna	2	Holy book	28.05.2020
	alekseeva	2	Holy book	01.02.2020
	alekseeva	2	Holy book	01.04.2020
	1changed	2	Holy book	01.05.2020
	1changed	2	Holy book	01.05.2020

На вкладке Клиенты можно посмотреть таблицу со всеми данными клиентов, а также добавить нового и изменить ФИ и паспортные данные у уже существующего. Для изменения ФИ и паспортных данных их нужно вводить парами, даже если часть не изменяется.

Библиотека К-района

Журнал Клиенты Книги Типы книг Статистика

ID	FIRST_NAME	LAST_NAME	PATHER_N
1	dasha	petrovna	Andreevna
2	marina	alekseeva	Vlad
3	dima	varf	Andreevich
21	1chanoed	1chanoed	OPath

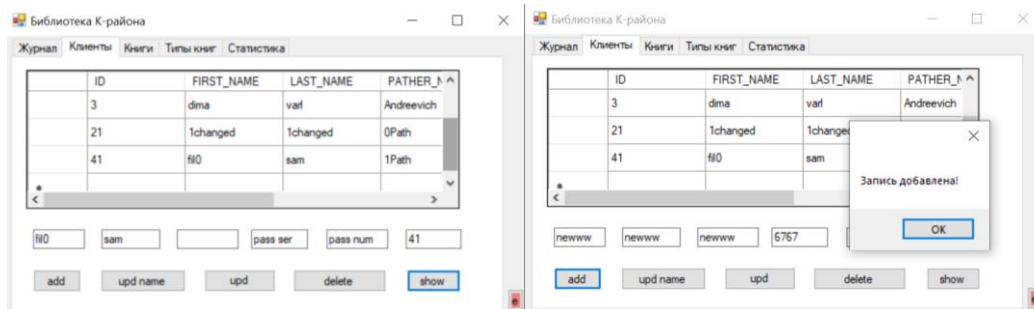
Библиотека К-района

Журнал Клиенты Книги Типы книг Статистика

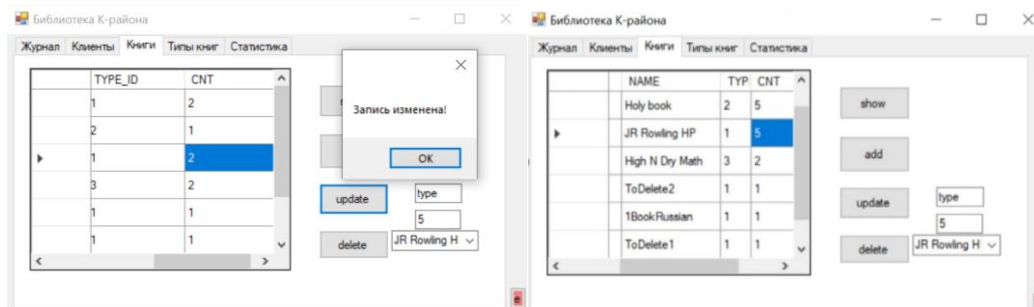
ID	FIRST_NAME	LAST_NAME	PATHER_N
3	dima	varf	Andreevich
21	1c		OPath
41	fl		1Path

Запись изменена!

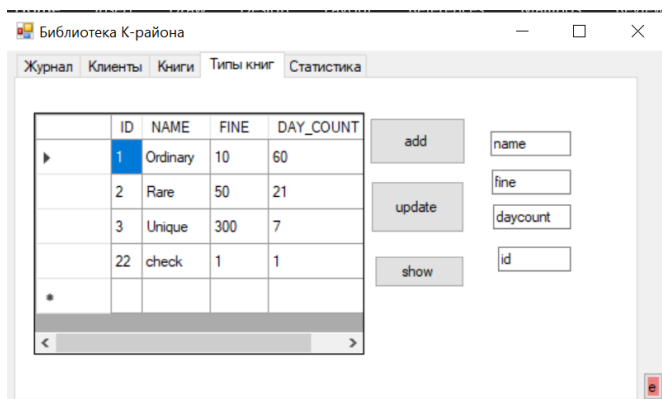
OK



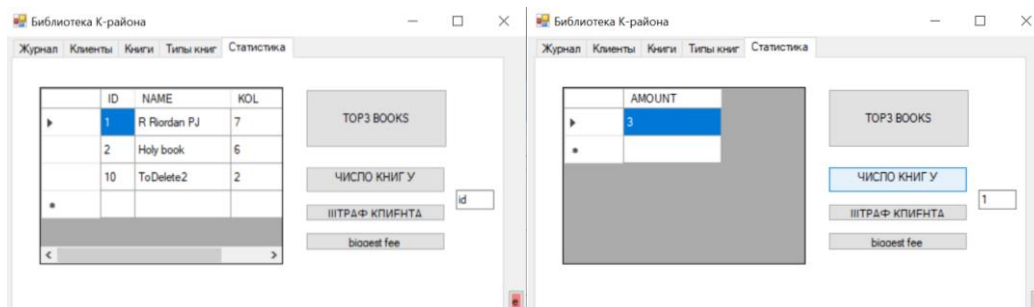
Во вкладке Книги можно посмотреть список всех книг, добавить новую, изменить количество существующих копий книги и удалить книгу.

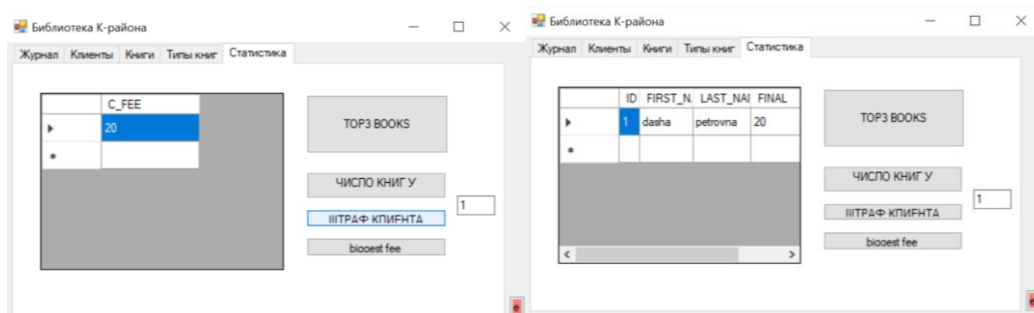


Во вкладке Типы книг можно посмотреть все типы книг, а также добавить новые или поменять размер штрафа у уже существующих.




Во вкладке Статистика можно посмотреть 3 самые популярные книги, число книг у заданного клиента и размер штрафа клиента, а также узнать размер самого большого штрафа.





Данные полученные во вкладке статистика записываются в отдельных текстовый файл, где можно их потом также прочитать:

 log – Блокнот

Файл Правка Формат Вид Справка

Взята книга 21.05.2020 0:00:00

3 Самые популярные книги:

1: R Riordan PJраз:4

2: Holy bookраз:4

3: ToDelete2раз:2

3 Самые популярные книги:

1: R Riordan PJ взята раз: 7

2: Holy book взята раз: 4

3: ToDelete2взята раз: 2

## **Вывод**

По теме автоматизация работы библиотека была разработана и реализована база данных.

В ходе работы пришлось столкнуться с проблемами поиска алгоритма для хеширования паролей и выбора платформы для реализации клиента.

Для хеширования выбран алгоритм md5 на основании простоты реализации. Выбор платформы обосновывался опытом работы с Windows Forms.

### **Список использованной литературы**

1. Кузнецов С. Д. Базы данных. Модели и языки. - Москва, Бином Пресс, 2008.
2. Herbert Schildt Java The Complete Reference, Seventh Edition. \_ McGraw-Hill Osborne
3. Beginning Oracle SQL  
Copyright © 2009 by Lex de Haan, Daniel Fink, Tim Gorman, Inger Jørgensen, Karen Morton
4. Oracle Database SQL Language Reference, 12c Release 1 (12.1)E17209-14  
Copyright © 1996, 2013, Oracle and/or its affiliates.  
Primary Author: Mary Beth Roeser